Enhancement of TextRank Algorithm using Coreference Resolution

Karla Jane Patosa[1], Marion James Hernandez[1], Vivien Agustin[1], Richard Regala[1], Khatalyn Mata[1], Dan Michael Cortez[1], Leisyl Mahusay[1], Andrew Bitancor[1], Perferinda Caubang[1]

[1] kjcpatosa2018@plm.edu.ph
[1] mjmhernandez2018@plm.edu.ph
[1] vaagustin@plm.edu.ph
[1] rcregala@plm.edu.ph
[1] kemata@plm.edu.ph
[1] dmacortez@plm.edu.ph
[1] lmocampo@plm.edu.ph
[1] agbitancor@plm.edu.ph
[1] ppcaubang@plm.edu.ph
[1] Computer Science Department, College of Engineering and Technology
Pamantasan ng Lungsod ng Maynila (University of the City of Manila)
Intramuros, Manila 1002, Philippines

Abstract

TextRank Algorithm is an unsupervised graph-based algorithm by Mihalcea with two primary applications, namely in text summarization and keyword extraction of a text document. This study will focus on the enhancement of the TextRank algorithm on the keyword extraction side. This paper introduces an enhanced version of the algorithm wherein the whole text document is preprocessed with a method known as Coreference Resolution, wherein this method normalizes every referenced entity in a text into a single entity. The application of this method to a document text with a longer sequence, outperforms the Precision, Recall, F1-Measure, and the Mean Average Precision metrics of the original algorithm.

Keywords: Natural Language Processing; Keyword Extraction; TextRank Algorithm; Coreference Resolution

1. Introduction

Keyword extraction (KE) can be described as the process of determining which lexical units best represent the document (Firoozeh et al., 2020). A lot of studies offer different approaches when extracting keywords. One of them is the use of machine learning methods and one of the most used is unsupervised learning. In this type of approach, keyword extraction methods are classified into three types: (a) statistical features, (b) topic models, and (c) graph models (Wang et al., 2020).

Keyword extraction based on statistical features produces subject keywords from candidates based on text analysis and statistics, as well as the computation of word frequency, probabilities, and other information extracted (Liu et al., 2020). One of the most used algorithms under the statistical model is the term frequency-inverse document frequency (Salton & Buckley, 1988). The document analysis is considered as a combination of topics in the keyword extraction methods based on the topic model since the chance that words emerge under

each topic is varied. As a result, once the document topics have been discovered, representative words from each topic represent the text's main information and can be used as keywords (Wang et al., 2020).

A graph-based ranking model is a method of determining the significance of a vertex within a graph (Mihalcea, 2004). In general, graph-based models see a text as a graph, with vertices representing words and edges representing the link between two words bound by the window size (Qiu et al., 2021). The basic concept being implemented here is a voting strategy to rank the vertices within a graph. When one vertex is linked to another, it is regarded as a vote for the other vertex. The higher the number of votes cast for a vertex, the higher the significance it will have (Wang et al., 2012, p. 545).

In 2004, Mihalcea and Tarau introduced a new graph-based ranking model which was based on Google's "PageRank" algorithm (Page et al., 1999). They named it "TextRank". This algorithm follows the same approach as PageRank but instead of scoring webpages, sentences or words are scored and ranked based on their importance in a given document. TextRank is an unsupervised approach for keyphrase extraction, which means that there is no need for training data.

TextRank has been used for a variety of tasks such as text summarization, keyword extraction, and sentence extraction. Through the years, several other papers tried to enhance this algorithm. This paper investigates the efficacy of introducing Coreference Resolution to the TextRank algorithm to be used specifically for keyword extraction.

2. Related Studies

KE is a textual information-processing task that automates the extraction of representative and characteristic terms from a document to express all of the document's key aspects. Natural language processing (NLP), a subfield of artificial intelligence (AI), is used in this technique to break down human language so that machines can understand and analyze it (Sutter, 2022). NLP allows for the development of solutions that allow for increasing levels of automation in the processing and analysis of qualitative data input (Heil et al., 2020). This aids with the problem of human bias by providing a more structured way of analyzing the responses.

According to Beliga et al., (2015), the goal of Keyword Extraction is to identify labels that provide the most relevant description of a document. Furthermore, the Keywords themselves that are to be extracted are independent of all corpora and can be used in others. In studying the terms that represent the most essential information included in the document, several nomenclatures are used: key phrases, key segments, key terms, or simply keywords. All of the synonyms provided have the same purpose, they characterize the topics discussed in a paper (Beliga, 2014). Keyword extraction has been utilized for different practical applications such as credibility assessment (Balcerzak et al., 2014), search engine optimization (Horasan, 2021), customer feedback/product review analysis (Wang & Zhang, 2017; Joung & Kim, 2021; Wang et al., 2018; Yang et al., 2019; Joung et al., 2018), and text categorization (Hulth & Megyesi, 2006; Hassaine et al., 2015; An & Chen, 2005). Also, multiple existing methods currently applied: Statistical, Linguistic, Machine Learning, Others (Ping-I and Shi-Jen, 2010, as cited in Beliga et al., 2015). The expanded version which included Linguistic and Others is according to Zhang et al.., 2008. Keyword extraction methods can be generally categorized as supervised or unsupervised (Beliga, 2014).

Turney (2000) and Frank et. al (1999) regarded KE as a supervised learning task. According to them, each term in a text is either a key phrase or not, and the challenge is appropriately categorizing each word into one of these two groups. Turney extracts keys using a genetic algorithm (GenEx) and a set of parametric heuristic criteria (Turney, 1999, as cited in Siddiqi & Sharan, 2015) while Frank et. al as well as Uzun (2005), extracts keys by using the Naive Bayesian classifier. Other algorithms implemented under supervised learning are: Support Vector Machines (SVM) classifier (Zhang et. al, 2006; Wu et al.,2009; Armouty & Tedmori, 2019; Guleria et al., 2021; Xu et al., 2010; Hasan et al., 2017), SVMRank (Cai & Cao, 2017), Least Square Support

Vector Machines (LS-SVM; Wu et al., 2007), C4.5 algorithm with features based on lexical chains (Ercan & Cicekli, 2007), a combination of KEA and lexical chains (Li & He, 2014), TF-IDF with Bi-gram expansion (Liu et al., 2008), k-means clustering with the cosine similarity distance between documents Vector Space Models (VSM; L'Huillier et al., 2010), and patent keyword extraction algorithm (PKEA; Hu et al., 2018). In most cases, the supervised method outperforms the unsupervised method (Kim & Kan, 2009, as cited in Sun et al., 2020). Articles, on the other hand, grow exponentially and alter over time, necessitating efficient and flexible key extraction. Supervised approaches require a document set with human-assigned key phrases (Liu et al., 2010), and this approach demands a significant amount of manual labor, thereby "reducing its utility for real-world application" (Kim & Kan, 2009).

Many studies offer keyword extraction techniques that employ either supervised or unsupervised learning (Hasan & Ng, 2014). In this paper, the focus will be on an unsupervised approach where a graph-based algorithm will be utilized.

Graph-based ranking algorithms are essentially a method of determining the significance of a vertex within a graph that is recursively based on global information derived from the entire graph The fundamental idea behind a graph-based ranking model is that of a "vote" or a "recommendation". When one vertex forms an edge to another vertex, it is essentially casting a vote for the other vertices. The more votes cast for a vertex, the more important the vertex is. Furthermore, the importance of the vote is determined by the vertex casting the vote. It is a result of the score associated with a vertex. and is determined by the votes cast for it and the score of the vertices casting these votes (Mihalcea and Tarau, 2004).

In 2004, Mihalcea and Tarau proposed a novel graph-based ranking model based on Google's "PageRank" algorithm (Page et al., 1999). It was given the name "TextRank" by the authors. This technique is similar to PageRank, only instead of scoring webpages, we score phrases or words and then order them according to their importance in a document. Several studies have attempted to improve this algorithm over the years to solve some of its shortcomings. Some use this algorithm as a basis for creating another novel algorithm.

ExpandRank (Wan and Xiao, 2008) is a TextRank extension that extracts keys using neighborhood knowledge. The authors of this study discuss the creation of a set of similar documents D for a given document to supply more knowledge. The goal of constructing a similar document set is to allow the model to use global information in addition to the local information found in any given document.

CollabRank organizes the documents into clusters and extracts the keywords from each cluster. The idea is that texts with similar themes contain similar keywords. There are two layers of keyword extraction. First, the words are ranked using a graph-based ranking method similar to PageRank (Page et al., 1998) at the cluster level. After that, the words are assessed on a document level by adding the saliency scores from each cluster. POS tags are used to identify eligible candidate keywords at the cluster level, and they are also used to assess whether the candidate key keywords are suitable (Timonen et al., 2012). CollabRank for collaborative single document keyphrase extraction takes advantage of the reciprocal influences across documents in the proper cluster context to improve the evaluation of the saliency of words and phrases (Wan & Xiao, 2008).

In 2017, Florescu and Caragea proposed a novel algorithm which they named "PositionRank. It is a fully unsupervised, graph-based model that computes a biased PageRank score for each candidate word by combining the position of words and their frequency in a document. The fundamental principle behind PositionRank is to give words that appear early in a document more weight (or probabilities) than terms that appear later in the document.

The EmbedRank algorithm (Bennani et al., 2018) is an unsupervised key extraction method that uses document embedding methods like Sent2Vec to encode both documents and candidate phrases as vectors in a continuous vector space. This research was the first to apply sentence-based vector representation algorithms for key extraction. By computing cosine similarity between the phrase vectors and the document vector, the vector representations aid in candidate ranking.

3. Existing TextRank Algorithm

3.1. Overview

Mihalcea and Tarau (2004) introduced a graph-based ranking algorithm that ranks the importance of vertices in a graph, borrowing some elements from other graph-based ranking algorithms, especially from the PageRank algorithm by Page et al. (1999), primarily its scoring of the vertices, which works on the principle of "voting" or "recommendation", that is, the score of a vertex is determined by its number of votes, and how "important" those votes are. This in essence makes the scoring formula recursive, as most graph-based algorithms are.

To be more formal, let $G = (V, E)$ represent the graph, where V is the set of all vertices in the graph and E is the set of all edges $\{(v, w) \in V \times V\}$ such that (v, w) is an ordered pair and $V \times V$ is the Cartesian product of the set V with itself. This shall be used to denote directed edges. A vertex V_i' score is calculated as follows (Brin and Page, 1998):

$$S(V_i) = (1 - d) + d * \sum_{j \in In(v_i)} \frac{1}{|Out(v_j)|} S(V_j) \qquad (1)$$

In the equation above, d is the Damping Factor which refers to the probability of a given vertex jumping into another vertex. This is usually set to 0.85 according to the PageRank algorithm (Page et al., 1998). $In(V_i)$ is the set of vertices that point to a particular vertex V_i (predecessors/ancestors), and $Out(V_j)$ is the set of vertices that vertex V_j points to (successors/children).

3.2. The Problem in TextRank Algorithm

Suppose the part-of-speech tagger is applied to each sentence in the document and suppose that there exists an expression in the document such that this expression refers to another expression in the document, it may either be forwards or backward of the initial expression. This relationship that involves the referents and the referee is called cataphora and anaphora. According to Jurafsky and Martin, (2022), anaphora is when the referring expression looks backward to the reference entity, while cataphora is first mentioned before the reference entity. Since the tagger only extracts words that are nouns and adjectives, this ignores the referring expressions that are usually pronouns, which decreases the score for the referred entity, since the algorithm only considers the co-occurring words filtered from the parts-of-speech tagger. This leads us to the problem that this paper is trying to address. TextRank only considers the local relationship. Zhou et al. (2022) stated that as a result, the extraction findings may be incomplete or inaccurate Qui and Zheng (2021) also stated that the drawbacks of the TextRank method (TM) originate from the fact that it only analyzes word co-occurrence and incipient word significance when extracting keywords. Since TextRank only considers local co-occurrences, there is no way of telling if two words are the same entity. The traditional TextRank originally removes stopwords. These are words that seem to be insignificant for keyword extraction including pronouns such as she, he, it, that, who, etc. But according to a study conducted by Basaldella et al. (2016), " The removal of such elements causes a loss of cohesion, both syntactically and semantically." According to them, the loss of syntactical and/or semantic information could occur if all pronouns are ignored without being replaced with a valuable substitute.

3.3. Pseudocode of TextRank Algorithm

```
Tokenization and Part-of-Speech-tagging
    For each word in the document:
        Split and store each word in an array and tag each word in its appropriate part of speech
Graph construction
    Initialize g = (v, e) as a graph data structure
    For each unit in the tokenized array:
        Store each unit into v
    For each vertex in v:
        Form an edge between two vertices that are co-occurrent within w units (words)
Scoring
    For each vertex in v: initialize the score associated with each vertex to 1
    Threshold = 0.0001
    iterations = 30
    For i in range(0, iterations):
        prev_score = summation(score)
        For each vertex in v:
            Apply the ws(v_i) formula (Equation 1 in Section 3.1) to each score
            associated with each vertex
        If prev_score - summation(score) <= threshold:
            Break
```

4. Enhanced TextRank Algorithm

4.1. Enhancement of the algorithm

To avoid the loss of cohesion in this problem, the importance of pronouns is taken into consideration. After the POS tagging part, coreference resolution will be applied to identify which words are referring to the same entity. After doing so, the next step would be to replace the pronouns with their referred entity. Consider the sentence, “Jane loves to sing. She is friends with James.” In this sentence the word she is referring to the word/name Jane. Therefore using the proposed solution, the word she will be replaced with Jane, therefore changing the sentences to “Jane loves to sing. Jane is friends with James.”

4.2. Pseudocode of the Enhanced Algorithm

```
Let M be a Pretrained Model for Coreference Resolution
resolved_documents = []
D = Dataset
for each d in D:
  resolved_documents.append(M.resolve_coreference(d))

for each d in resolved_documents:
  TextRank(d)
```

5. Methodology

The researchers employed an experimental research design to investigate whether the proposed solution will enhance the original TextRank algorithm. To determine if the proposed change to the algorithm enhances the existing TextRank algorithm used for keyword extraction, the original algorithm will serve as the baseline, with the enhanced algorithm with its proposed changes being compared to it by using the Performance Metrics defined in Section 4.2. Two datasets will be used for producing the results of the experiments in which the baseline algorithm and the enhanced algorithm will have their performance compared to see if an enhancement did take place. The two datasets are from the SemEval-2010 Task 5 by Kim et al. (2010) and from Marujo et al. (2012). For both datasets, both the baseline and the enhanced algorithm are evaluated on their respective test sets. The SemEval-2010 Task 5, consists of 100 scientific articles, each varying from 6 to 8 pages. While the dataset from Marujo et al. (2012) consists of 50 news stories. As can be seen, the SemEval-2010 Task 5 dataset is longer in its input sequence and number of documents. Both the test sets of the two datasets contain annotated keywords, which shall be considered as the ground truth. To simplify the process of matching the extracted keyword from running the baseline algorithm and the enhanced algorithm on both the datasets, only the exact match with the annotated keyword is considered.

5.1. Coreference Resolution

Merging all the linked entities into a single entity is also known as resolving the coreferences. The document is normalized by resolving all the known coreferences. Two state-of-the-art models are used, namely the neuralcoref model from spacy which is based on the papers of Manning and Clark (2016), and the Coreference Resolution model from AllenNLP which is based on the paper of Lee et al. (2018).

5.2. Performance Metrics

According to Jiang, et al. (2009), the traditional problem of keyword extraction is often formally considered a classification problem, in which a model is used to classify whether a word is a keyword or not. Since the TextRank algorithm ranks the extracted keywords on their score, together with the traditional metrics used for Keyword Extraction, we consider another additional metric called Mean Average Precision or MAP for short. The definitions and notations used are from Manning et al., (2008) and Jiang, et al., (2009).

$$\frac{|\{ground\ truth\ keywords \cap extracted\ keywords\}|}{\{extracted\ keywords\}} \quad (2)$$

$$\frac{|\{ground\ truth\ keywords \cap extracted\ keywords\}|}{\{ground\ truth\ keywords\}} \quad (3)$$

$$2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

$$Mean \sum_{1}^{Q} \sum_{1}^{k} Average\ Precision(k) * rel(k) \quad (5)$$

Precision (Equation 2) shows how good the measure on how good the extracted results are. The true keywords can all be extracted, but with many irrelevant keywords, which lowers the precision. Equation 3 calculates Recall, which measures if every extracted keyword is the true keyword. Equation 4 calculates the F1 Measure, which is the accuracy of the algorithm for extraction. To see how well our algorithm works based on the ranking problem, we take the top k keywords for each measure above. Additionally, we introduce another metric, the Mean Average Precision (Equation 5). We represent the query Q as the document d as follows: For each document d in dataset D, we take the mean of the average precision of each document. That is, we compute the summation of all the Precision of the top k keywords, where k ranges from 1 to min(|extracted keywords|, |true keywords|) multiplied by a relevance function rel(k), wherein it measures if the kth keyword in the list of extracted keywords is in the set of true keywords. Then it is divided by the total size of the true keywords associated with each document.

6. Results

The performance metrics of the original algorithm and the various enhancements are presented here. For the SemEval 2010 dataset, specifically the coreference resolution enhancement, only the neuralcoref model will be used to resolve the coreferences due to the limitations of the resources of the researchers.

Table 1 Baseline Algorithm

Dataset	Precision	Recall	F1-Measure	Mean Average Precision
SemEval 2010	3.08	19.61	5.33	1.13
Marujo et al. (2012)	23.86	19.70	21.58	6.66

Table 2 With Coreference Resolution

Dataset	Precision	Recall	F1-Measure	Mean Average Precision
SemEval 2010 (neuralcoref)	5.00	15.20	7.52	1.58
Marujo et al. (2012) (neuralcoref)	23.35	19.31	21.14	5.94
Marujo et al. (2012) (AllenNLP)	22.18	17.53	19.58	5.00

The bigger dataset SemEval 2010 showed improvements in all metrics apart from its recall, due to it merging every linked entity into one single entity. Since keyword extraction here is treated as a ranking problem, the researchers have managed to successfully show that the application of resolving coreferences improves the scoring of the top-ranked keywords extracted. The total accuracy from the F1 Measure is also shown to improve even when the recall is reduced.

The dataset from Marujo et al., (2012) meanwhile showed a steady decrease the more accurate the model that is used. This is a smaller dataset and hence it collapses the linked entities into one, which massively impacts the recall, and since the scoring of the multiwords is not normalized. Applying some of the heuristics and features from Marujo et al. (2012) and Basaldella et al. (2016) to normalize the scoring of the phrases should increase its performance metrics even more since the coreference resolution model that the researchers used are even more powerful.

7. Conclusion

From the results collected, the enhanced algorithm performs more accurately on documents with longer sequences, and also with bigger datasets. While the AllenNLP model is more accurate, it is computationally expensive since it consumes a lot of memory and the use of a GPU might be needed to make use of this in documents with longer sequences. Meanwhile, true to its word, the neuralcoref implementation by spacy truly is for production use and can be applied efficiently.

For future works, the researchers recommend considering more features like noun phrase chunking to properly extract multiword expressions. Also, the researchers plan on implementing contextual embedding and semantic similarity measures to further improve the accuracy of the TextRank algorithm.

Acknowledgments

The researchers wish to express their gratitude to God for providing them with the knowledge and wisdom they need throughout their journey. To their family for their unwavering love and support. To Prof. Vivien Agustin, for being the study's adviser. For the continuous support and guidance from the Pamantasan Ng Lungsod Ng Maynila - College of Engineering and Technology - Computer Science Department faculty and staff.

References

An, J., & Chen, Y. P. (2005, May). Keyword extraction for text categorization. In Proceedings of the 2005 International Conference on Active Media Technology, 2005.(AMT 2005). (pp. 556-561). IEEE.

Armouty, B., & Tedmori, S. (2019, April). Automated keyword extraction using support vector machine from Arabic news documents. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 342-346). IEEE.

Balcerzak, B., Jaworski, W., & Wierzbicki, A. (2014, August). Application of TextRank algorithm for credibility assessment. In 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) (Vol. 1, pp. 451-454). IEEE.

Basaldella, M., Chiaradia, G., & Tasso, C. (2016, December). Evaluating anaphora and coreference resolution to improve automatic keyphrase extraction. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 804-814).

Beliga, S. (2014). Keyword extraction: a review of methods and approaches. University of Rijeka, Department of Informatics, Rijeka, 1(9).

Beliga, S., Meštrović, A., Martinčić-Ipšić, S. (2015). An Overview of Graph-Based Keyword Extraction Methods and Approaches. Journal of Information and Organizational Sciences 39(1):1-20

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., & Jaggi, M. (2018). Simple unsupervised keyphrase extraction using sentence embeddings. arXiv preprint arXiv:1801.04470.

Cai, X., & Cao, S. (2017, August). A keyword extraction method based on learning to rank. In 2017 13th International Conference on Semantics, Knowledge and Grids (SKG) (pp. 194-197). IEEE.

Ercan, G., & Cicekli, I. (2007). Using lexical chains for keyword extraction. Information Processing & Management, 43(6), 1705-1714.

Firoozeh, N., Nazarenko, A., Alizon, F., & Daille, B. (2020). Keyword extraction: Issues and methods. Natural Language Engineering, 26(3), 259-291.

Florescu, C., & Caragea, C. (2017, July). Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1105-1115).

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-Specific Keyphrase Extraction. JCAI'99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 668-673.

Guleria, A., Sood, R., & Singh, P. (2021). Automatic Keyphrase Extraction Using SVM. In Advances in Communication and Computational Technology (pp. 945-956). Springer, Singapore.

Hasan, H. M., Sanyal, F., Chaki, D., & Ali, M. H. (2017, October). An empirical study of important keyword extraction techniques from documents. In 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM) (pp. 91-94). IEEE.

Hasan, K. S., & Ng, V. (2014, June). Automatic keyphrase extraction: A survey of the state of the art. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1262-1273).

Hassaine, A., Mecheter, S., & Jaoua, A. (2015, September). Text categorization using hyper rectangular keyword extraction: Application to news articles classification. In International conference on relational and algebraic methods in computer science (pp. 312-325). Springer, Cham.

Heil, D., Jenset, G., & McGillivray, B. (2020). Extracting Keywords from Open-Ended Business Survey Questions. Journal of Data Mining & Digital Humanities, 2020.

HORASAN, F. (2021). Keyword extraction for search engine optimization using latent semantic analysis. Politeknik Dergisi, 24(2), 473-479.

Hu, J., Li, S., Yao, Y., Yu, L., Yang, G., & Hu, J. (2018). Patent keyword extraction algorithm based on distributed representation for patent classification. Entropy, 20(2), 104.

Jiang, X., Hu, Y., Li, H., (2009). A ranking approach to keyword extraction. SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 756-757).

Joung, J., & Kim, H. M. (2021). Automated Keyword Filtering in Latent Dirichlet Allocation for Identifying Product Attributes From Online Reviews. Journal of Mechanical Design, 143(8).

Jurafsky, D., & Martin, J. (2022). Speech and Language Processing (3rd edition draft). https://web.stanford.edu/~jurafsky/slp3/

Kim, S. N., & Kan, M. Y. (2009, August). Re-examining automatic keyphrase extraction approaches in scientific articles. In Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009) (pp. 9-16).

Kim, S. N., Medelyan, O., Kan, M., & Baldwin, T. (2010, July). SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles. SemEval '10: Proceedings of the 5th International Workshop on Semantic Evaluation (pp. 21- 26).

Lee, K., & He, L., & Zettlemoyer, L. (2018). Higher-order Coreference Resolution with Coarse-to-fine Inference. NAACL 2018.

L'Huillier, G., Hevia, A., Weber, R., & Rios, S. (2010, May). Latent semantic analysis and keyword extraction for phishing classification. In 2010 IEEE international conference on intelligence and security informatics (pp. 129-131). IEEE.

Li, Z., & He, B. (2014, September). Adding Lexical Chain to Keyphrase Extraction. In 2014 11th Web Information System and Application Conference (pp. 254-257). IEEE.

Liu, F., Huang, X., Huang, W., & Duan, S. X. (2020). Performance evaluation of keyword extraction methods and visualization for student online comments. Symmetry, 12(11), 1923.

Liu, F., Liu, F., & Liu, Y. (2008, December). Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion. In 2008 IEEE Spoken Language Technology Workshop (pp. 181-184). IEEE.

Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010, October). Automatic keyphrase extraction via topic decomposition. In Proceedings of the 2010 conference on empirical methods in natural language processing (pp. 366-376).

Manning, C., & Clark, K. (2016). Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In Association for Computational Linguistics (ACL).

Marujo, L., Gershman, A., Carbonell, J., Frederking, R., & Neto, J. (2012). Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization. In 8th International Conference on Language Resources and Evaluation (LREC 2012).

Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab, 1999-66.

Qiu, D., & Zheng, Q. (2021). Improving TextRank Algorithm for Automatic Keyword Extraction with Tolerance Rough Set. International Journal of Fuzzy Systems, 1-11.

Qiu, Q., Xie, Z., Xie, H., & Wang, B. (2021). GKEEP: An Enhanced Graph-Based Keyword Extractor With Error-Feedback Propagation for Geoscience Reports. Earth and Space Science, 8(5), e2020EA001602.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.

Siddiqi, S., & Sharan, A. (2015). Keyword and keyphrase extraction techniques: a literature review. International Journal of Computer Applications, 109(2).

Sutter, R. D. (2022, January 5). Mastering NLP: A guide to keyword extraction. Radix. Retrieved April 11, 2022, from https://radix.ai/blog/2022/1/mastering-nlp-a-guide-to-keyword-extraction/

Timonen, M., Toivanen, T., Teng, Y., Chen, C., & He, L. (2012, October). Informativeness-based Keyword Extraction from Short Documents. In KDIR (pp. 411-421).

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. Information retrieval, 2(4), 303-336.

Uzun, Y. (2005). Keyword extraction using naive bayes. In Bilkent University, Department of Computer Science, Turkey www. cs. bilkent. edu. tr/~ guvenir/courses/CS550/Workshop/Yasin_Uzun. pdf.

Wan, X., & Xiao, J. (2008, August). CollabRank: towards a collaborative approach to single-document keyphrase extraction. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008) (pp. 969-976).

Wan, X., & Xiao, J. (2008, July). Single document keyphrase extraction using neighborhood knowledge. In AAAI (Vol. 8, pp. 855-860).

Wang, H., Ye, J., Yu, Z., Wang, J., & Mao, C. (2020). Unsupervised keyword extraction methods based on a word graph network. International Journal of Ambient Computing and Intelligence (IJACI), 11(2), 68-79.

Wang, W. L., Lei, J., Zhiguo, G., & Luo, X. (Eds.). (2012). Web Information Systems and Mining: International Conference, WISM 2012, Chengdu, China, October 26-28, 2012, Proceedings (Vol. 7529). Springer.

Wang, Y., & Zhang, J. (2017, December). Keyword extraction from online product reviews based on bi-directional LSTM recurrent neural network. In 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 2241-2245). IEEE.

Wang, Y., Mo, D. Y., & Tseng, M. M. (2018). Mapping customer needs to design parameters in the front end of product design by applying deep learning. CIRP Annals, 67(1), 145-148.

Wu, C., Marchese, M., Jiang, J., Ivanyukovich, A., & Liang, Y. (2007). Machine Learning-Based Keywords Extraction for Scientific Literature. J. Univers. Comput. Sci., 13(10), 1471-1483.

Wu, C., Marchese, M., Jiang, J., Ivanyukovich, A., & Liang, Y. (2007). Machine Learning-Based Keywords Extraction for Scientific Literature. J. Univers. Comput. Sci., 13(10), 1471-1483.

Xu, S., Yang, S., & Lau, F. (2010, July). Keyword extraction and headline generation using novel word features. In Twenty-Fourth AAAI Conference on Artificial Intelligence.

Zhang, K., Xu, H., Tang, J., & Li, J. (2006, June). Keyword extraction using support vector machine. In international conference on web-age information management (pp. 85-96). Springer, Berlin, Heidelberg.

Zhou, N., Shi, W., Liang, R., & Zhong, N. (2022). TextRank Keyword Extraction Algorithm Using Word Vector Clustering Based on Rough Data-Deduction. Computational Intelligence and Neuroscience, 2022.